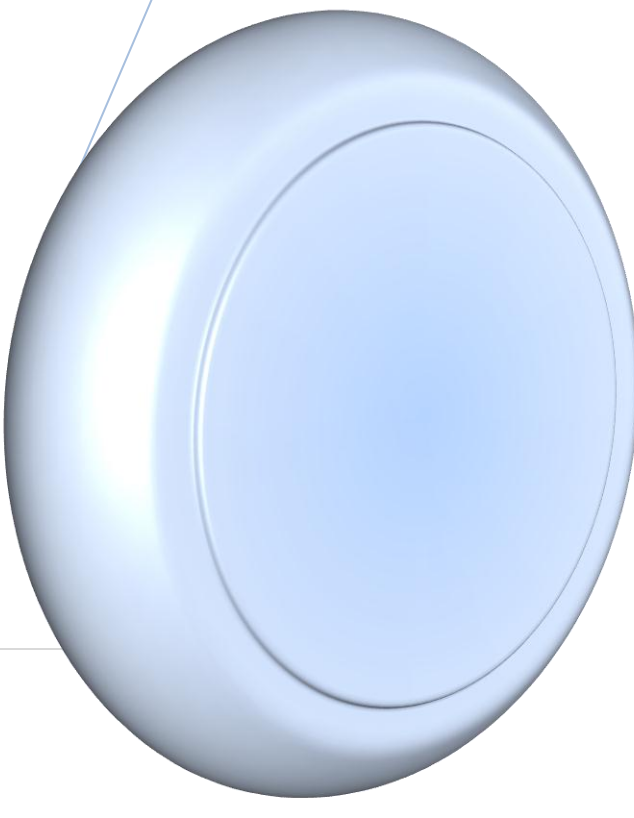
[Type the document subtitle]

Created by Colin Greene – x08399573



Supervisor – Paul Hayes
5/1/2012

Declaration Cover Sheet for BSHSS4 Project Submission

SECTION 1 *Student to complete*

Name: **Colin Greene**
Student ID: **X08399573**
Supervisor: **Paul Hayes**

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature:_____ Date:_____

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the college's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

Complete the sections above and attach it to the front of one of the copies of your assignment,

What constitutes plagiarism or cheating?

The following is extracted from the college's formal statement on plagiarism as quoted in the Student Handbooks. References to "assignments" should be taken to include any piece of work submitted for assessment.

Paraphrasing refers to taking the ideas, words or work of another, putting it into your own words and crediting the source. This is acceptable academic practice provided you ensure that credit is given to the author. Plagiarism refers to copying the ideas and work of another and misrepresenting it as your own. This is completely unacceptable and is prohibited in all academic institutions. It is a serious offence and may result in a fail grade and/or disciplinary action. All sources that you use in your writing must be acknowledged and included in the reference or bibliography section. If a particular piece of writing proves difficult to paraphrase, or you want to include it in its original form, it must be enclosed in quotation marks and credit given to the author.

When referring to the work of another author within the text of your project you must give the author's surname and the date the work was published. Full details for each source must then be given in the bibliography at the end of the project

Penalties for Plagiarism

If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the college's Disciplinary Committee. Where the Disciplinary Committee makes a finding that there has been plagiarism, the Disciplinary Committee may recommend

- that a student's marks shall be reduced
- that the student be deemed not to have passed the assignment
- that other forms of assessment undertaken in that academic year by the same student be declared void
- that other examinations sat by the same student at the same sitting be declared void

Further penalties are also possible including

- suspending a student college for a specified time,
- expelling a student from college,
- prohibiting a student from sitting any examination or assessment.,
- the imposition of a fine and
- the requirement that a student to attend additional or other lectures or courses or undertake additional academic work.

Table of Contents

1. Executive Summary....4
2. Introduction....5
2.1 Background....5
2.2 Home Replacement Applications in Development and Available Today....6
2.3 Aim....7
2.4 Technology Overview....8
3. System....7
3.1 Structure of Technical Report....8
3.2 Requirements....9
3.2.1 Functional Requirements....10
3.2.2 User Requirements....11
3.2.3 Environmental Requirements....11
3.2.4 Interface Requirements....12
3.3 Design and Architecture....12
....13
3.3.1 Context Diagram....14
....14
....14
3.3.2 Use Case Diagrams....14
3.4 Implementation – Technologies....16
3.4.1 Overview....16
3.4.2 Eclipse IDE/ Android Development Tools (ADT)....16
3.4.2.1 Installation of Eclipse IDE and ADT....16
3.4.3 Development of the CanvasLauncher Home Replacement Application....19
3.4.3.1 Developing an Android Project in Eclipse....19
3. 5 Testing....22
3.5.3.1 Survey....22
3.6 Results....23
3.6.1 Issues....23
3.6.2 Failed Functionality's....23
3.7 Conclusion....24
4. The Future/Evolution....24
5. Bibliography....25

Appendix 1: Project Proposal……26

Appendix 2: Requirements Specification ……30

Appendix 3: Code Segments That Were Unusable…….. 42

Appendix 4: Survey……46

Appendix 5: Reflective Journals……50

1. Executive Summary

Due to the introduction of the open source android platform application development has become more and more common throughout the android consumer base. People now have the ability to develop their own applications and with this the levels of diversity and innovation for the android platform have increased dramatically.

This report presents one of these innovative and diverse concepts that have been gathering more interest from developers and consumers, the *"CanvasLauncher",* a home replacement application which allows users to improve the user interface and functionality of one's mobile device without tampering with the phone in a non-sanctioned way, i.e. rooting. This essentially means it's a safe and fully endorsed way of giving ones phone a more customized feel and look and is more accessible to those who are not as technically minded as others.

Outlined in this report is the:

1. Reasoning behind this applications development.
2. What if provides for its users.
3. The technologies implemented during its development.
4. The benefits it holds for those who will ultimately use it.

2. Introduction

In this section we will outline the initial idea and concept of the CanvasLauncher Home Replacement Application along with the aim of the project and the technologies used to develop the system.

2.1 Background

Due to the introduction of the open source android platform application development has become more and more common throughout the android consumer base. People now have the ability to develop their own applications and with this the levels of diversity and innovation for the android platform have increased dramatically.

This level of development has created a pool of developers all creating and sharing their own applications with the rest of the android consumer base via the android market, this has allowed ideas to be created and developed further via developer forums, online tutorials and online developer blogs. Such levels of interaction between developers has allowed android to boast some highly innovative applications that have widespread market appeal. One of these innovations has been in the development of "*Home Screen Replacement Applications*", more commonly referred to in the android community as "*Android Launchers*".

"*Launcher is the name given to the part of the Android user interface that lets users customize the home screen (e.g. the phone's desktop), launch mobile apps, make phone calls, and perform other tasks on Android devices (devices that use the Android mobile operating system).*"

(Webopedia Home Launcher, 2012)

All android phones come with their own stock launcher which has been installed on the device during its development, for every brand of phone there is a different stock launcher. This can be easily seen when comparing the Sony Ericsson Xperia mobile phone range with the HTC mobile phone range, as shown below in figure 1.0.

Fugure 1.0 - Sony Ericsson Stock Launcher vs HTC Stock Launcher

As seen above the graphical user interface (GUI) on both phones is very different, with the introduction of" *Home Replacement Applications"* any phones GUI can be changed to suit the users preference. This means that any android device can be made to emulate that of another device, or any other theme or combination of display settings which that particular home replacement application allows. For this reason home replacement applications are becoming more and more popular.

2.2 Home Replacement Applications in Development and Available Today

Upon the beginning of this project to my knowledge 20+ home replacement applications were available to android users via the android market. Since then however development in this area has increased dramatically, today in the range of 60+ android launchers are available with many others still in development.

Examples of such launchers in development are:

1. ROU Launcher
2. Slick UI Launcher

Some of the other launchers that finished development during the course of the development of this project are:

1. TSF Shell
2. Nova Launcher
3. ICS Launcher

Some well-established launchers available on the android market are:

1. Go Launcher Ex – 10million+ downloads
2. Launcher Pro – 5million+ downloads
3. ADW Launcher - 5million+ downloads
4. Launcher 7 – 1million+ downloads
5. Regina 3D Launcher - 1million + downloads

This shows how much of an interest there now is in developing such applications and how well received they have been by android users up to now.

This area of android development is heating up with each set of developers trying to establish new innovative features that will highlight their launcher above the others to the android consumer base. For example GO launcher ex boasts a wide range of custom applications that can only be used with the installation of the launcher, LauncherPro boasts a wide range of themes and user interface preferences for those users who want to fully modify the look of their device. This level of competition is proving to make android launcher development a very innovative and exciting area to be a part of in today's world.

2.3 Aim

This section outlines the aims of this project and what will hopefully be achieved during the course of its development.

The main aim of this project is to develop a working android home replacement application which allows users to customize the look and feel of their android device to suit their own personal preferences. The overall outcome hoped to be achieved can be broken down into several areas.

1. Home
    - Support multiple home screens
    - Render Widgets
2. An Application Picker

3. Preference screen
4. Custom Widgets.

The application should be able to be installed on a variety of android devices allowing it to be used across a wider user market than if it was brand specific.

2.4 Technology Overview

The approach taken in the development of my android home replacement application was through the use of the Eclipse IDE in conjunction with the ADT (Android Development Tools) plugin. This is designed to give a powerful development environment in which to build a wide range of android applications.

The ADT extends the basic capabilities of the Eclipse IDE which in turn allows users to quickly set up and develop android projects in various ways such as;

1. Application User Interface
2. Debugging and cleaning of the project
3. Exporting of the application .apk for distribution online
4. Provides an emulator to fully test your application
5. Allows easy installation of your application on any android device for further testing
6. Online tutorials and support

3. System

In this section we will outline the structure of this technical report by presenting a detailed description of the requirements for the android home replacement application. Next we look at the design and architecture for the system and how the several components of the application come together to form the overall working application. Upon establishing these we will look at the technologies used and implemented in the system. Next we will describe the testing and evaluation process that took place over the course of the projects development and finally we will finish up on the future prospects and evolution of the CanvasLauncher which will show the way in which application can be further developed beyond the point of this project.

3.1 Structure of Technical Report

The structure of this technical report is designed to help describe in the most effective and efficient manner the development process that was undertaken during the course of this project. The introduction above gave a brief description of the project as a whole, why it was chosen as my project of choice and an overview of the android environment in regard to the popularity and rise in interest in home screen replacement application development.

The rest of this paper will be structured in the following way;

1) Section 3.2

   - Here will discuss the requirements for the *CanvasLauncher home replacement application* that need to be addressed for the applications development.

` 2) Section 3.3

   - Here we will discuss the design and architecture of the *Canvas Launcher home replacement application.* This section will show and explain the architecture of an android application along with presenting a set of various diagrams created during the initial stages of the development process.

3) Section 3.4

   - In this section we shall take a look at the technologies used and implement during the development process. Along with this a guide to the best use of these technologies will also be provided.

4) Section 3.5

   - In this section the testing for the *CanvasLauncher home screen replacement application* will be outlined and discussed.

5) Section 3.6

   - In this section we will look at the overall result of the development process in terms of what was actually achieved and why.

6) Section 3.7

   - In this section we will look at the overall Conclusions of the development process.

7) Section 4

   - Here we will look at the future development opportunities of the CanvasLauncher Home Replacement Application and the possible evolution of the system.

3.2 Requirements

This area covers the process of determining what the systems should be able to do from a number of different perspectives. For this reason it is one of the first processes to be carried out in the Software Development Lifecycle as it helps to ensure that the projects vital components and requirements are

catered for and integrated into the project plan. These requirements will fall under one of several headings from the following list;

1. Functional Requirements
2. User Requirements
3. Environmental Requirements
4. Interface Requirements

In order to properly assess the various requirements for this project I explored the various developer forums for answers in relation to the development of a fully functioning home replacement application. In conjunction with this the various dev. teams specialising in this area of android development were also appropriately questioned and studied as in order to improve upon what's already there one must go direct to its source to establish and figure out how it was developed and implemented.

3.2.1 Functional Requirements

In this section we will establish the functional requirements of the CanvasLauncher home replacement application. A functional requirement at its simplest is essentially something that the system must do and perform from the perspective of the end user interacting with the application.

The functional requirements for the CanvasLauncher are;

1. ***Ability To Replace The Stock Android Launcher***
   This is the most essential requirement as without the application being able to fully replace the stock home launcher the application would not be an efficient home replacement app. This requirement will be activated by the user after the application has been installed. Once installed the application will be ready to run, upon its installation the user will have initialised the application to replace the stock launcher running on the android device.

2. ***User Interaction via Touch Screen***
   This requirement is the bases upon which the user interacts and uses the application, without the application being responsive to the users commands via the touch screen the application will be unusable as it will be unresponsive to user requests. This requirement will be used constantly by the user once the home replacement application has been installed on the android enabled device. The CanvasLauncher will be the way in which the user interacts with their phone to select application such as messaging, email etc. from now on so it is a vitally important function to the applications overall success.

3. ***Ability To Launch Applications Installed On The Android Device***
   As the application is a home replacement app it utilises all the applications installed on the android device and allows the user to continue to use them through its own interface. After the launcher has been installed and is running on the device the user will be shown the applications installed on the phone via the application drawer. Upon selecting an application (for example messages) the CanvasLauncher will begin to run that application normally like the stock launcher would.

4. ***Ability To Allow Users To Customize The Look of Their Phone***
   With allowing users to customize their phone in ways that are not allowed through the stock launcher there is an incentive for using the *CanvasLauncher* as your home replacement application. The customization features will be accessible via the launcher settings window. Here the user will be able to edit and customize their phone with an array of options.

3.2.2 User Requirements

In this section we will establish the user requirements of the *CanvasLauncher home replacement application*. Such requirements are based on outlining the required behaviour from the user interacting with the system and the expected output from the system to the user.

1. ***User Interaction (Request)***
   The user will interact with the application via the mobile device touch screen. This in turn will identify the specific request of the user based on where they have pressed on screen.

2. ***User Response***
   The device will graphically display on screen the necessary output to the user based on the users request interaction.

3. ***CanvasLauncher User Manual***
   In addition to the CanvasLauncher Home Replacement Application being developed a user manual will also be developed. This will provide users with all the required information, both written and visual which will be needed to install and fully use the application.

3.2.3 Environmental Requirements

In this section we will establish the environmental requirements of the *CanvasLauncher home replacement application*. These requirements deal specifically with the ideal environment needed in order for the system to work as expected.

1. ***Physical Environment***
   The CanvasLauncher Home Replacement Application will be developed primarily on a Sony Vaio laptop with a Intel(R) Pentium(R) processor , 4GB of RAM and a 455GB Hard Drive. All I will be working with is this personal laptop upon which the appropriate development software will be installed.

   - Due to the risk off overheating which may leave the development laptop unusable or there being a loss of data I will be keeping it, along with all saved data in a cool area that allows for sufficient ventilation of the systems.

- Due to the installation requirements of the ADT I will need;
    1. Windows 7 (32/64 bit)
    2. The Eclipse IDE
    3. JDK 6
    4. 2GB Disk Space for all installations

3.2.4 Interface Requirements

In this section we will establish the interface requirements of the CanvasLauncher home replacement application.

1. ***Mobile Device***
   The user interacts with the application via the GUI of the android device that allows input from the user and then displays an output based on what the user has requested. This GUI will display all menu screens, icons, applications, widgets etc. it is essentially the central hub of all the applications output which is to be displayed to the user.

3.3 Design and Architecture

In this section we will look at the systems architecture and design from a high level overview first, and then we look at the associated diagrams that help display the system in a more in depth and detailed manner.

The architecture of the CanvasLauncher Home Replacement Application clearly defines and outlines each of the individual segments that ultimately make up the application as a whole.

The CanvasLauncher's architecture consists of the following;

1. ***An Android Device***
    - Put simply this will be the mobile device upon which the CanvasLauncher Home Replacement Application will run and operate.

2. ***The Application Framework***
    - Due to androids open source development platform developers have the ability to build new exciting and innovative applications. The framework API's available to developers are the exact same as those used for the core applications on all android devices, due to this components are easily reusable and allow for the capabilities of specific applications to be used again by other applications.
3. ***Libraries***

- The android framework consists of multiple libraries which allow for a wide range of capabilities on the android system. These capabilities are provided to users via the android framework. Some of them are;

    i. **System C library**

    ii. **Media Libraries**

    iii. **SGL**

4. Linux Kernel

- This provides core system services such as security, memory management, process management, network stack, and driver mode.

Below is a diagram which shows the applications architecture:

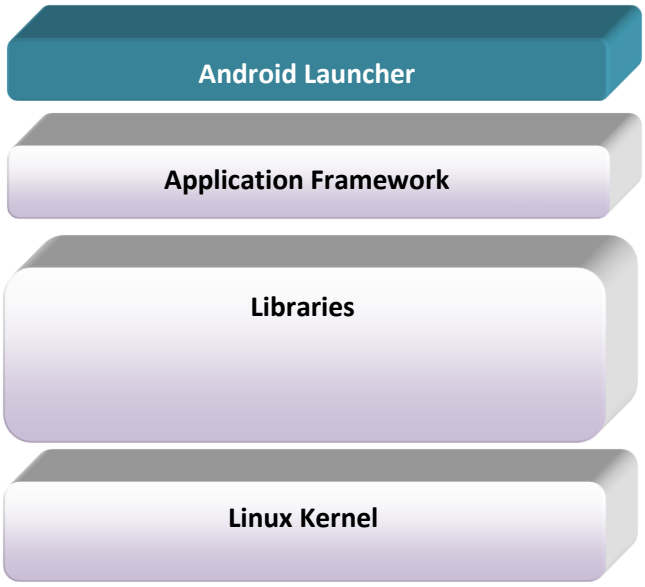
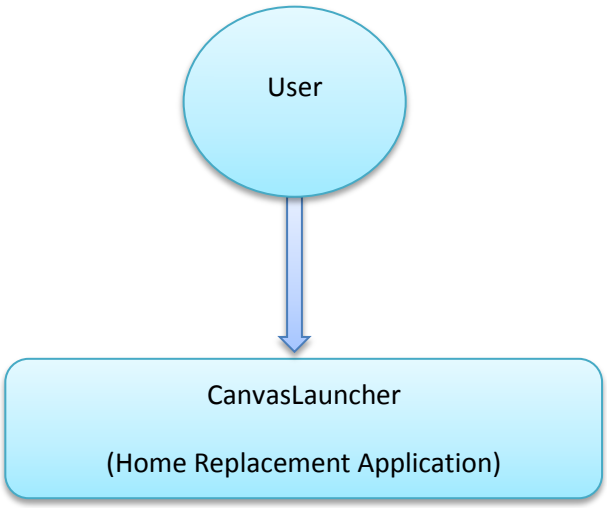


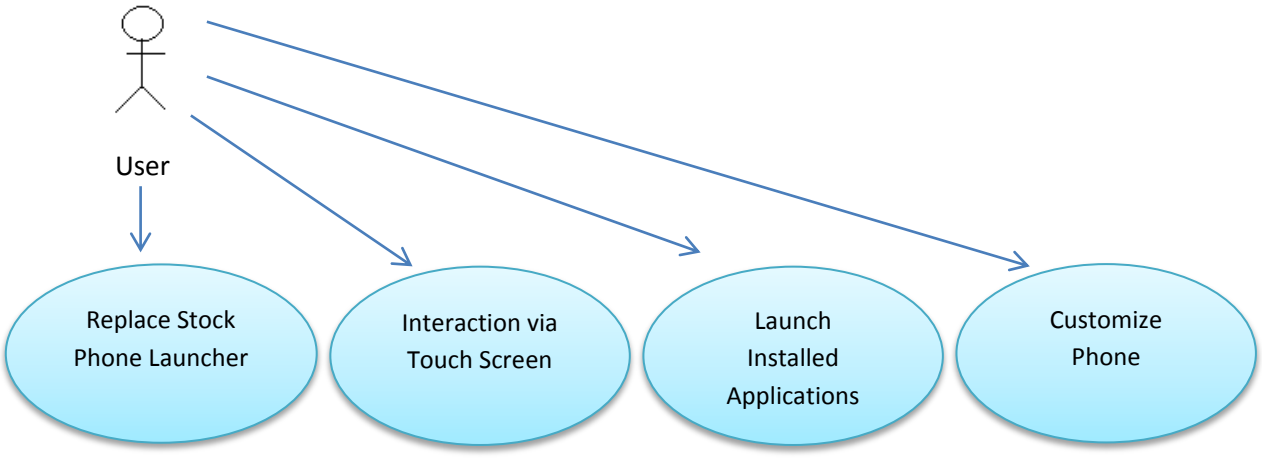
Figure 2.0 – System Architecture

3.3.1 Context Diagram

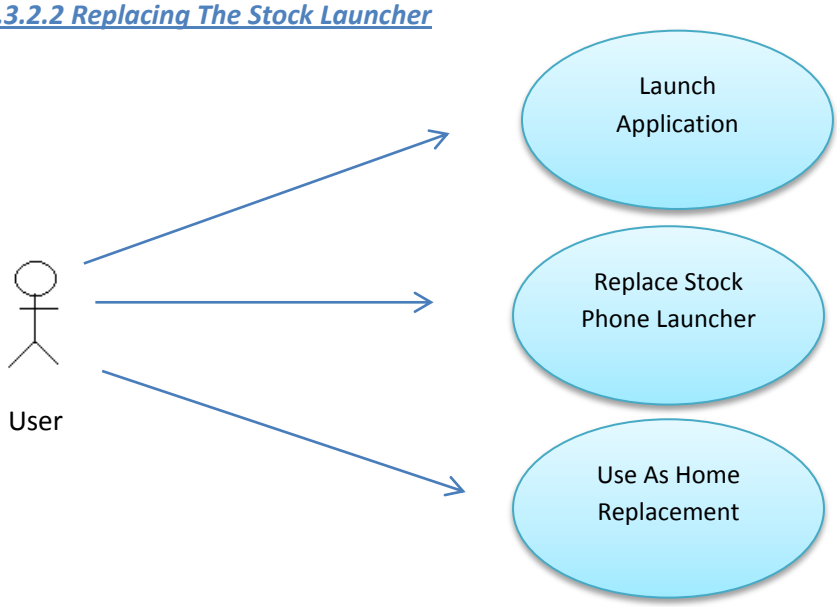


3.3.2 Use Case Diagrams

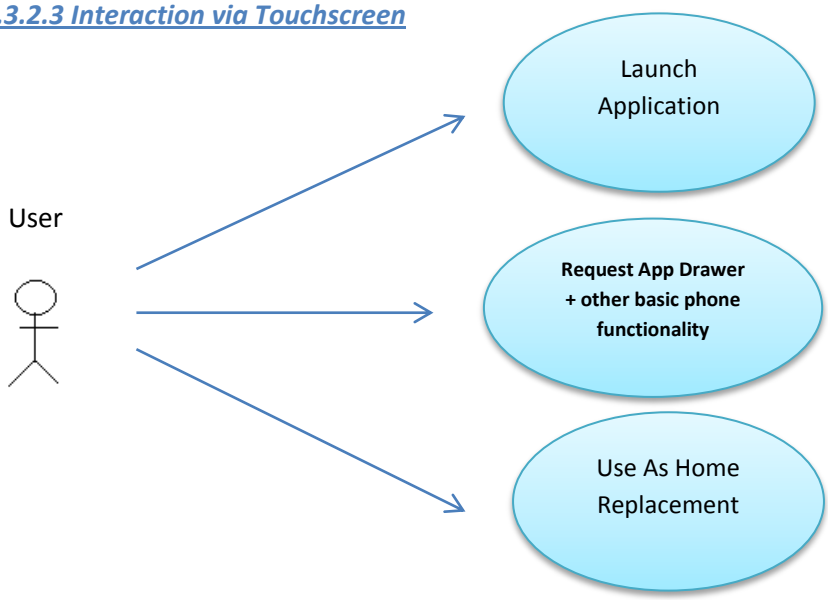
3.3.2.1 Android Launcher Overview



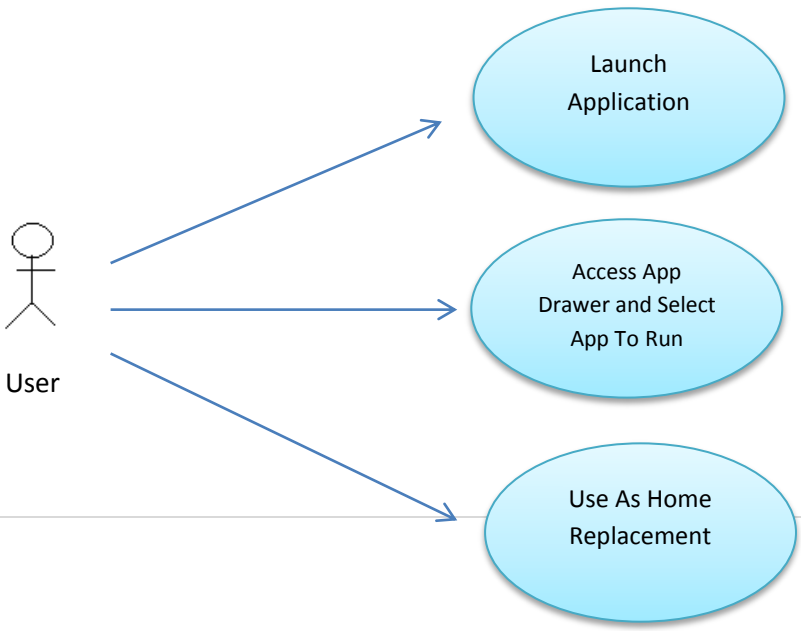
3.3.2.2 Replacing The Stock Launcher



3.3.2.3 Interaction via Touchscreen



3.3.2.4 Launch Installed Phone Applications



3.4 Implementation – Technologies

The purpose of the following section is to highlight the technologies used in the development and implementation of the *CanvasLauncher Home Replacement Application*.

Also in this section you will find a guide outlining the steps involved in the installation and setup of the development environment used for this project.

3.4.1 Overview

The approach taken in the development of my android home replacement application was through the use of the Eclipse IDE in conjunction with the ADT (Android Development Tools) plugin. This is designed to give a powerful development environment in which to build a wide range of android applications.

The ADT extends the basic capabilities of the Eclipse IDE which in turn allows users to quickly set up and develop android projects in various ways such as;

1. Application User Interface
2. Debugging and cleaning of the project
3. Exporting of the application .apk for distribution online
4. Provides an emulator to fully test your application
5. Allows easy installation of your application on any android device for further testing
6. Online tutorials and support

3.4.2 Eclipse IDE/ Android Development Tools (ADT)

Eclipse is an environment that boasts a wide array of plug-ins which enables it to support multiple languages for all kinds of development projects. For the purpose of this project the Google developed framework for the development of applications for Android s was used, more commonly known as ADT or Android Development Tools.

The ADT plugin for Eclipse allows users to create fully functioning android applications due to its extensive libraries, built in emulator and security features. It is for these reasons that the ADT plug in for Eclipse is the most popular choice for android development.

3.4.2.1 Installation of Eclipse IDE and ADT

This section will show in a step by step format the setup of the environment for the development of the CanvasLauncher Home Replacement Application.

Step 1 – Download the Eclipse IDE and Android Development Tools to the development PC

- Downloading The Eclipse IDE

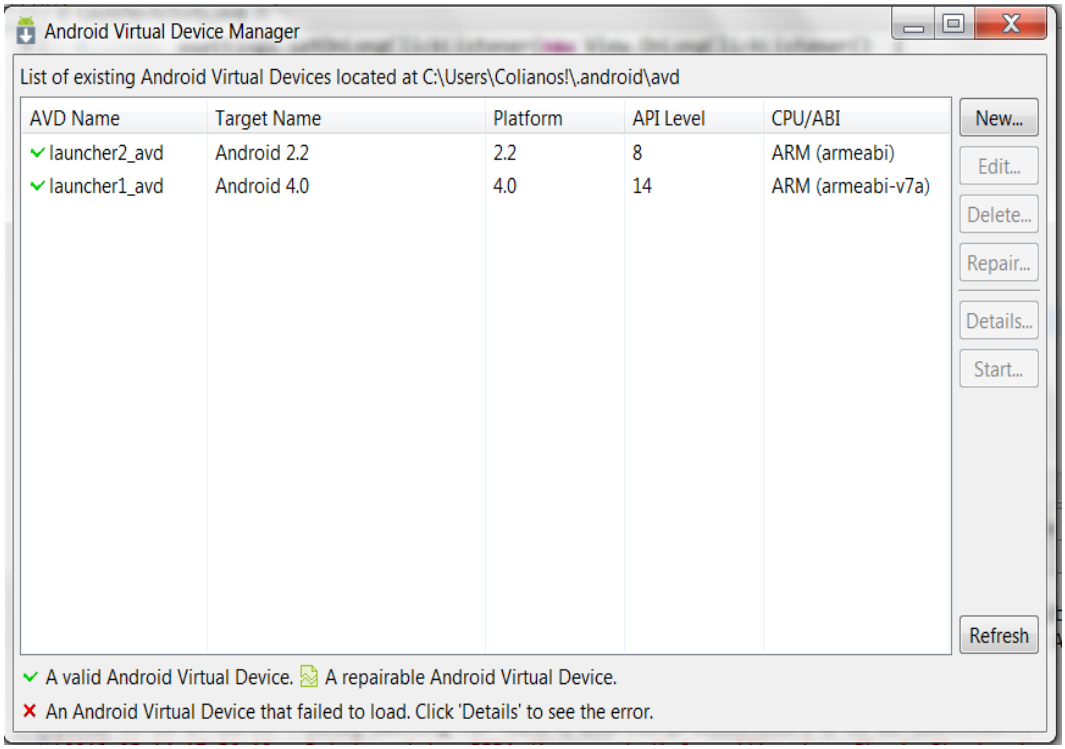
- Download the Eclipse IDE zip file(eclipse-SDK-3.7.2-win32-x86_64.zip) from this link: http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops/R-3.7.2-201202080800/eclipse-SDK-3.7.2-win32-x86_64.zip
- Extract the files from the zip file just downloaded to a specific location. This will create the eclipse folder containing all you need to run the Eclipse IDE.
- Open this extracted folder and run the Eclipse Application.

- Downloading The Android SDK
  - Download the Android SDK zip file(android-sdk_r18-windows.zip) from the following location: http://developer.android.com/sdk/index.html
  - Extract the contents of this zip file once downloaded to a specific location on your development PC.

- Download and install the ADT plugin for the Eclipse IDE
  - Firstly you must run Eclipse.
  - Once up and running Select Help > Install New Software.
  - You will be presented with a window to enter a URL of your desired software download location.
  - Enter the following URL into this allotted space and select OK once complete https://dl-ssl.google.com/android/eclipse/
  - Select the checkbox next to Developer Tools.
  - Click “Next”.
  - Click “Next” and “Finish”.
  - Once complete you must then restart Eclipse in order to activate the new downloaded ADT plugin.
- Setting Up The ADT plugin
  - Run the Eclipse IDE as you normally would.
  - Select Window > Preferences.
  - Now you must select Browse and locate the Android SDK you previously installed in the location of your choice.
  - Once located click “Apply” and “OK”.

Step 2 – Setting up the Android Virtual Device using The AVD Manager

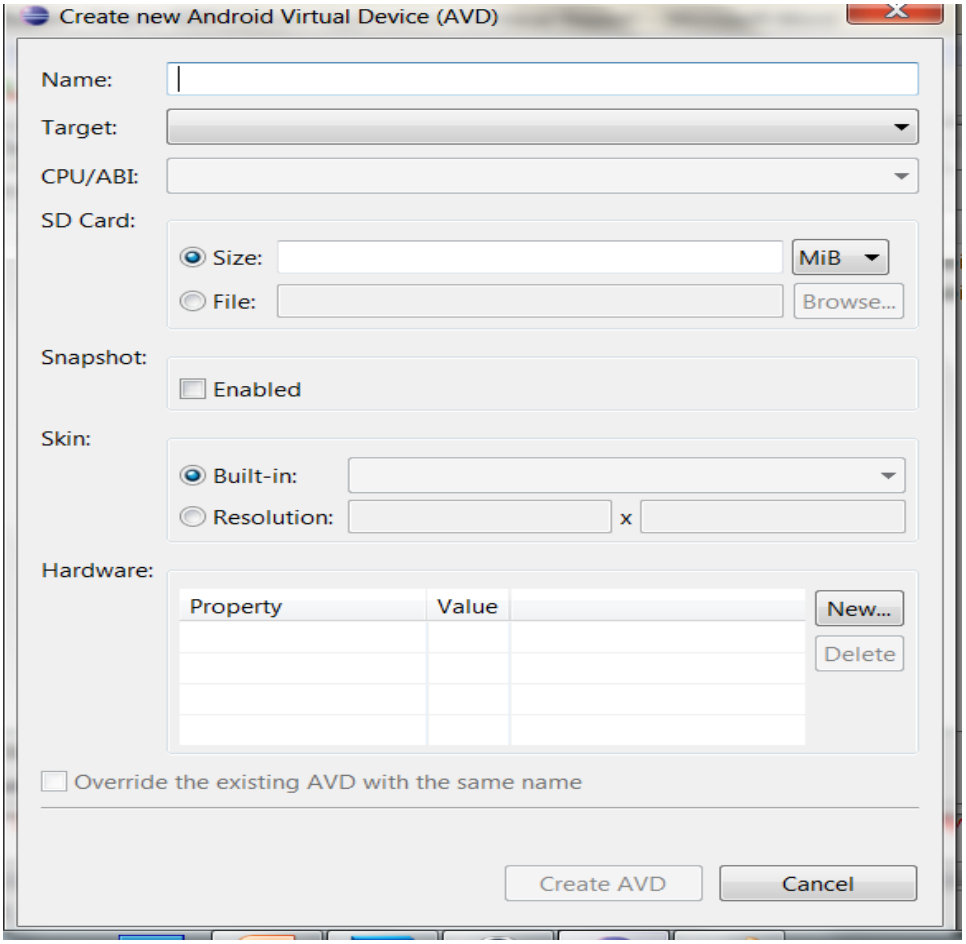
The Android Virtual Device (AVD) is a tool that allows users to model different android device configurations and assign them to the built in Android Emulator. This means an android application built with android 2.2 will run on a different AVD than an android application built with android 4.0.



- In order to create an AVD first you must start the AVD Manager in Eclipse.
  - To do this simply select Window > AVD Manager, this will open up the current list of AVD's available to you(As seen in the above diagram).

- Select "New" and proceed to enter in the following details;

  - Name
  - Platform Target(android 2.2 or android 4.0)
  - An SD Card size if needed
  - Additional hardware features can be added if required for your desired project



- Once complete select “Create AVD”
- Your AVD has now been created and is ready for use with you android applications.

3.4.3 Development of the CanvasLauncher Home Replacement Application

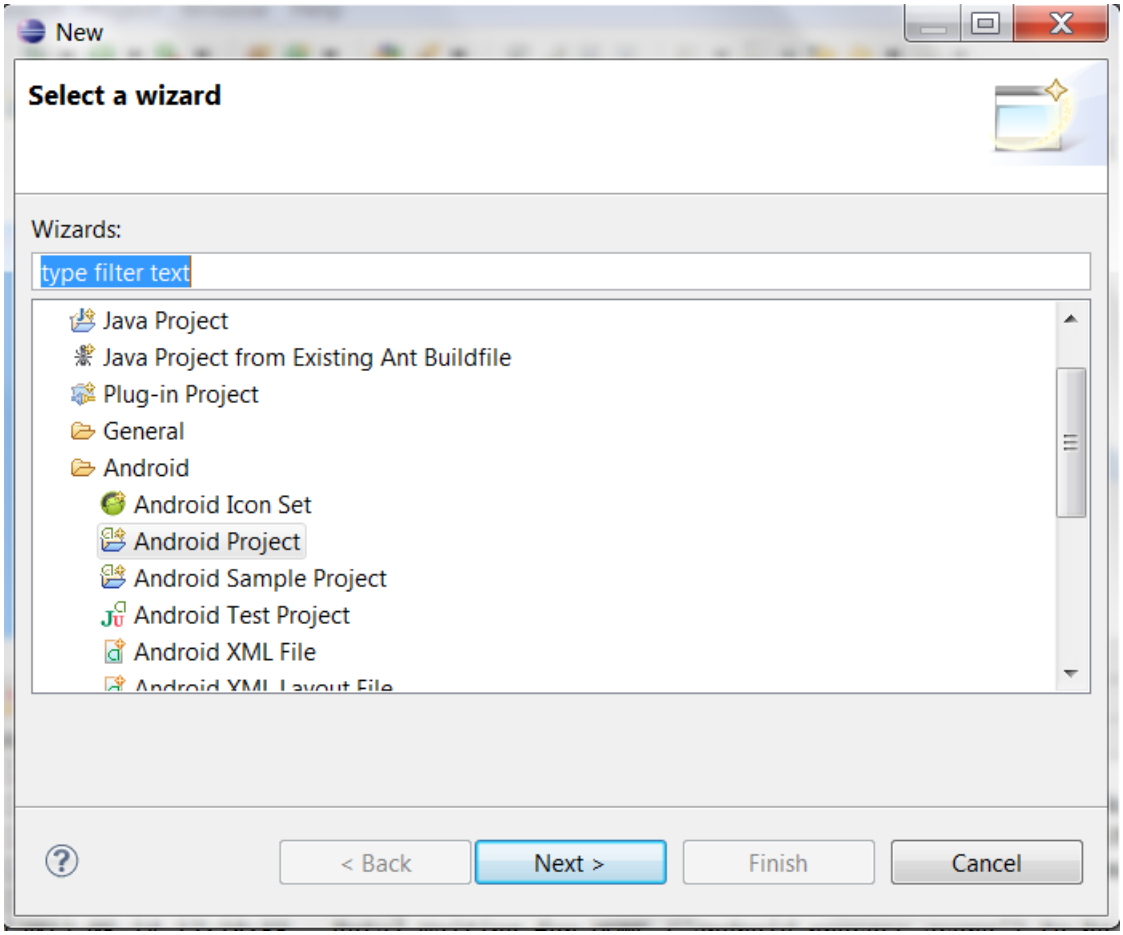
In this section will outline the process followed in the development of the CanvasLauncher Home Replacement Application.

3.4.3.1 Developing an Android Project in Eclipse

Step 1- Creating an Android Project

- Run Eclipse and wait for it to load

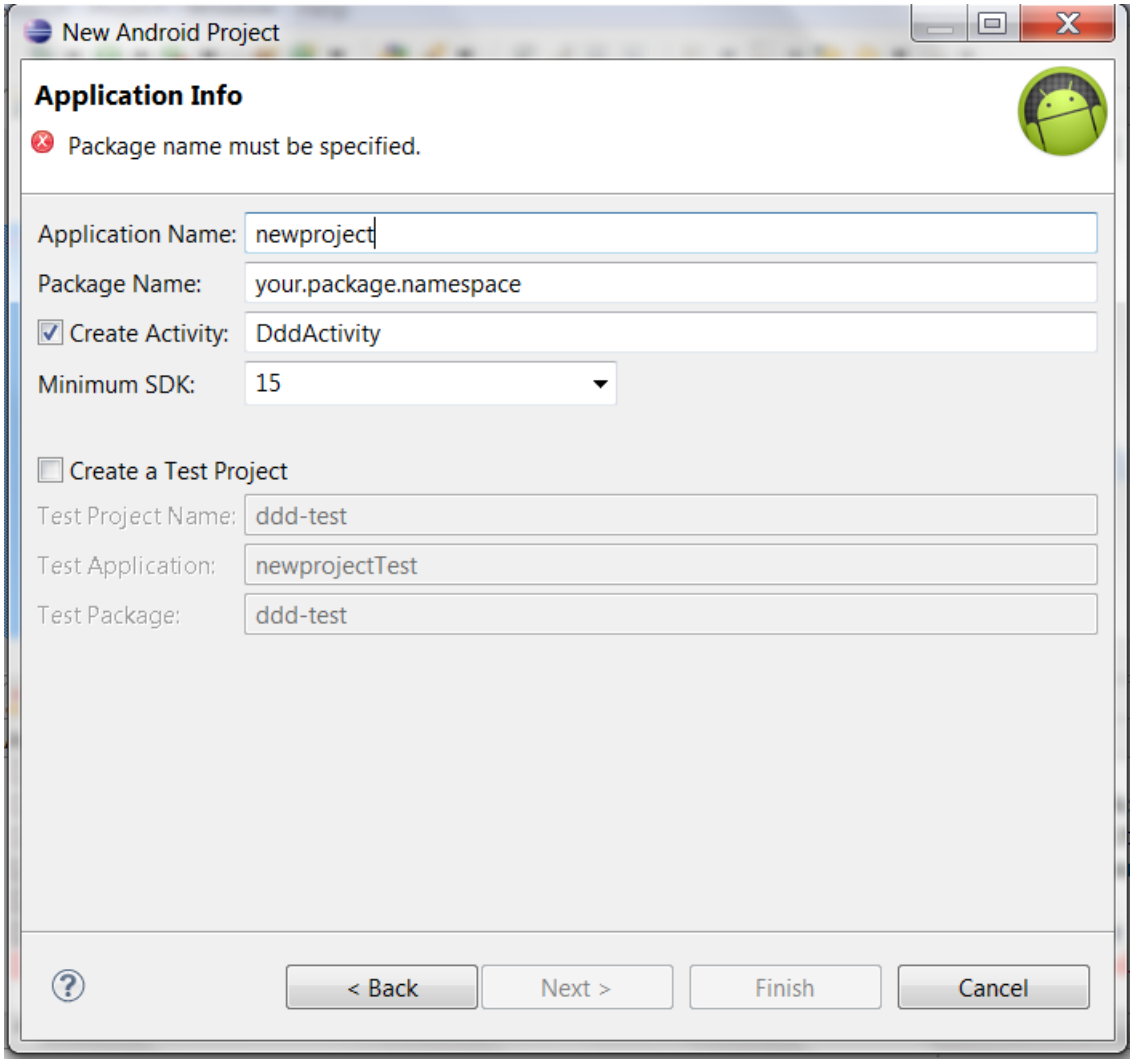
- Select File > New and then Other from the proceeding menu list.



- Select Android Project and click “Next”

- Enter in the details of your new android project

  - Project Name
  - Package Name
  - Android Platform Target
  - Main Activity Class Name



- Once complete click “Finish”
- Your android project has now been set up and is ready to be developed.

3.4.4 Other Technologies Used

Other technologies were used in conjunction with the tools used to develop the functionality of the project. These technologies were implemented for the development of the GUI, Icons and overall display of the application.

The technologies used to perform these tasks were:

1. Adobe Fireworks
2. Adobe Photoshop
3. Paint
4. DroidDraw

Each of these technologies proved useful during the development of the CanvasLauncher Home Replacement Application.

3. 5 Testing

Testing is a vital stage in the development of any software application. Testing involves ensuring that your application is acting correctly and performing as expected under various situations which in turn determines whether or not the system is meeting the requirements set out for it at the beginning of its development.

3.5.1 Unit Testing

Testing was carried out throughout the course of the projects development upon each stage of significant progress. This form of testing is referred to as Unit Testing. This involves testing each of the individual elements of the system as they are developed to ensure they are working as expected. This ensures that the individual components that will ultimately make the up the finished system work efficiently, however it does not necessarily ensure that these components when combined will give the system its full functionality as further errors and problems may occur.

3.5.2 System Testing

Upon the completion of the project System Testing began, this form of testing begins once the system has been developed as far as possible and is being integrated for use by users. It essentially involves testing the system as a whole to ensure what was set out at the beginning of the project in terms of the system requirements have been met. The system testing for the CanvasLauncher Home Replacement Application began upon the launchers completion and installation from Eclipse to a self-efficient android mobile device e.g. the Samsung galaxy Ace.

3.5.3 User Acceptance Testing

The final stage of testing allowed potential users of the application to use and view the application. This was done either on a display device provided by myself or on their own android device by installing the .apk file. This was carried out to gather feedback from the potential user base in relation to their thoughts on the look/feel of the application along with its functionality. This type of testing is referred to as User Acceptance Testing.

3.5.3.1 Survey

A simple survey was given to each of those individuals who interacted and used the application during the user acceptance testing phase. The results and findings of the user acceptance testing where gathered and evaluated, which in turn allowed for the assessment of the overall end user opinion of the CanvasLauncher Home Replacement Application. This survey was distributed online via the website http://www.surveymonkey.com. The survey used can be viewed at the following URL: http://www.surveymonkey.com/s/8MYHLWW

The survey consists of 7 simple questions designed to ensure the user can answer it as quickly and effectively as possible. Each question relates to various areas and the overall appeal of the application.

3.6 Results

In this section we will look at the overall results of the *CanvasLauncher Home Replacement Application.*

The final result at the end of the development process of the CanvasLauncher Home Replacement Application highlighted the extensive work needed in order to fully develop and realise a competent and efficient android launcher.

The end result over the course of the development process accurately portrayed the overall lack of resources to guide one in the development of an android home replacement application, unlike traditional applications which has plenty of source material to begin development with tutorials explaining the process behind their development.

3.6.1 Issues

This can be seen due to the number of issues that arose during the development process. These issues arose due to a significant lack of online or any other forms of resources including tutorials, guides, and books etc on the topic of home screen replacement applications.

The overall result reflects this due to the overall project concluding with several features being un resolved and incomplete constituting their exclusion from the end product. Also those features which did work when combined together created a number of errors meaning they are currently still independent applications with the hopes of them being merged together in the near future still hopeful.

3.6.2 Failed Functionality's

Over the course of the development process several attempts were made to develop and create various forms of functionality for the CanvasLauncher Home Replacement Application. These various features did not make it into the final project due to their unfinished state but they can be viewed in Section 3 of the appendices for reference in relation to the issues that arose and difficulties found during development.

3.7 Conclusion

This report has guided you through the development process of the CanvasLauncher Home Replacement Application from its initial concept in terms of what functionality it hoped to perform and the benefits it in turn provides the user with up to the inevitable reality of what was done to achieve this and the result achieved through these methods.

Overall this process was enjoyable and introduced the android platform to me from a developer's point of view, whereas before I only had experience with android from a consumer point of view. Seeing what goes into producing an android application has given me a new found sense of appreciation for the thousands of developers who provide the android market with new applications on a daily basis.

The main and in fact only disadvantage during the development of this project was in the lack of online resources available on the topic of "home replacement applications". The initial tutorials available do provide some insight into certain aspects of the application but there are no comprehensive guides, tools etc available to developers for this type of development. For this reason it was a fairly slow and tedious process which ultimately has led to some features being unimplemented into the final version of the project as was hoped at the beginning(although some features are working as their own independent applications).

4. The Future/Evolution

With the growth of the android platform being an ever increasing entity I do feel that further development of this project for its introduction into the mass market should be done as the consumer base is there for it. Users of the android platform are always looking for new ways to customize their devices. This can be seen in the rate of developers setting up dev. group projects linked to launcher development. The rate in which this topic has gathered more and more interest in recent time's shows users and developers are ready to establish new and innovative ways for users to interact with the android platform. This ultimately means that it is inevitable until the development process and guides of such applications becomes available to those interested in beginning development in this area.

5. Bibliography

- Android Development Team. (2012). *Home.* Available: http://developer.android.com/index.html. Last accessed 26th April 2012.

- Elance Blog. (2010). *The 10 Best Android Resources on the Web.* Available: https://www.elance.com/p/blog/2010/05/the_10_best_android_resources_on_the_web.html. Last accessed 10th November 2012.

- Paranoid Android. (2011). *Android Development.* Available: http://andytsui.wordpress.com/. Last accessed 12th May 2012.

- SurveyMonkey. (2011). Available: http://www.surveymonkey.com. Last accessed 14th May 2012.

- Various Authors. (2012). *Top Questions.* Available: http://stackoverflow.com/. Last accessed 20th April 2012.

- Webopedia, (2012). *Android Launcher.* Available: http://www.webopedia.com/TERM/A/android_launcher.html. Last accessed 12th May 2012.

Appendix 1: Project Proposal

Software Systems Project 2011
Android Launcher

Name: Colin Greene

Student Number: 08399573

E-mail: greene_colin@yahoo.ie

BSc (Hons) in Software Systems

Date : 27/09/2011

Objectives

The main objective of this project is to develop an android launcher which can be used as a home replacement for the stock android interface that comes with all android smartphones. Many of these stock user interfaces are rigid and lack the ability to allow user's full customization of the look and feel of their phone.

With my launcher I aim to address this problem which in turn will allow the user to customize their phone in way such as app icon appearance, widgets, layouts, animations, labels and transitions.

This launcher will be developed with the aim of having a nice easy to use interface that is simple to look at and understand as it will be potentially used by people with a wide range of backgrounds and technological abilities, with some being much less used to the capabilities of android smartphones.

Background

The reason I chose to do a android launcher for my project was due to the fact that as a android user myself I have never come across a launcher that has met each of my own personal needs when it comes to customizing the look and feel of my phone. Most launchers allow one or two cool features in the area of appearence customization, I want to create a launcher that is heavily focused in this area which will provide android users like myself with the perfect template for creating a 100% custom designed interface on their phones

Also as a android user I have always wanted to create an application that addresses the problems I myself found while using those available on the android market. This project will allow me to do just that and will stand to me in the long term with the android platform becoming much more pridominately used in todays society.

Currently there are very few launchers available to the public and even fewer that are free, for this reason I see this area in android development as one which has not yet been exploited by the development community fully and one which allows for alot of innovation.

Technical Approach

After coming up with the initial idea of creating a home replacement app (android launcher) I began to research and gather information on the best way to go about building the application as efficiently as possible.

Due to my project being solely for the android platform there was no need to worry about it being operational on a number of platforms. Therefore I began researching the android development website on the fundamentals of android development along with external online resources based more specifically on android launcher design and fundamental functionality.

This research led me to conclusion that the project would be written in java and built using the android SDK for windows development as the computer I will be using runs solely on windows. Finally I would need a development kit in which I would write and compile my application, I decided to use Eclipse as it was recommended by android and is an open source development toolkit

meaning it would be fully compatible with the android SDK and is a toolkit I have experience using in the past.

Special Resources Required

To write and code my project properly and to ensure no steps in the development process are missed I will first need to familiarize myself with android development. I will initially begin this process by reading online resources, learning the coding standards and doing simple tutorials to ensure everything is set up correctly on my development pc.

Other resources will vary over time from books, websites, forums and lecturer advice. The bulk of the material I will use as a reference for developing my android launcher will come from online tutorials and videos.

Project Plan

Attached to zip file uploaded on Moodle

Technical Details

This project will be written in java using the android SDK for windows and done so using eclipse as my development software. The main part of the application will be to allow users to install the launcher on their device, launch it which will in turn replace the stock home screen interface (not by default but temporarily until the launcher is closed down) and then allow them to customize the layout of their device as you would with any other home replacement app.

Evaluation

The android development kit comes with an emulator which will allow me to view the initial look and functionality of the application throughout its development, this is a great advantage to have as I won't have to repeatedly install it on my android device to test it. However after each step/task has been complete I will attempt to view the app on an android device to make sure the layout and design translates well from the pc to phone.

The majority of my class will be able to help me evaluate the current state of the application after each initial stage as they themselves have android phones and are interested in android development. Another advantage of this fact is the variety in android phones available to me with their permission to test the application on.

Consultation 1

Paul Stynes

I went to Paul with my initial idea of creating an android launcher, after telling him what it was and the fundamental concept behind what I am trying to create he told me to make sure I flesh out my idea into the several key aspects that will make up my project and to talk to Paul Hayes about it due to it being on the android platform.

Consultation 2

Michael Bradford

After talking to Michael about my project idea he encouraged me to research in depth the process of creating a launcher and to find sample code which could be used as a starting point in understanding and developing my launcher and how it works. He told me to check Google code and to ensure I put a substantial amount of time into specifying the functional and non-functional requirements of the project.

Also he stressed to stick to the objectives laid out from the start to ensure I don't get distracted and stray of course in the development process.

Proposed Supervisor

I would like to put down Michael Bradford and Paul Hayes as a potential supervisor for my project. I haven't yet talked to Paul Hayes about this matter as I was unable to reach him but I was told he has a keen interest in android development meaning he may be the most suitable to supervise the development of my android launcher.

Any other lecturers who have an interest in android development that would be willing to supervise my project would be a better match than someone who is new to this area. If possible I would like to work with someone with a knowledge base in android development as I believe it would ensure a smoother development of the project.

_____

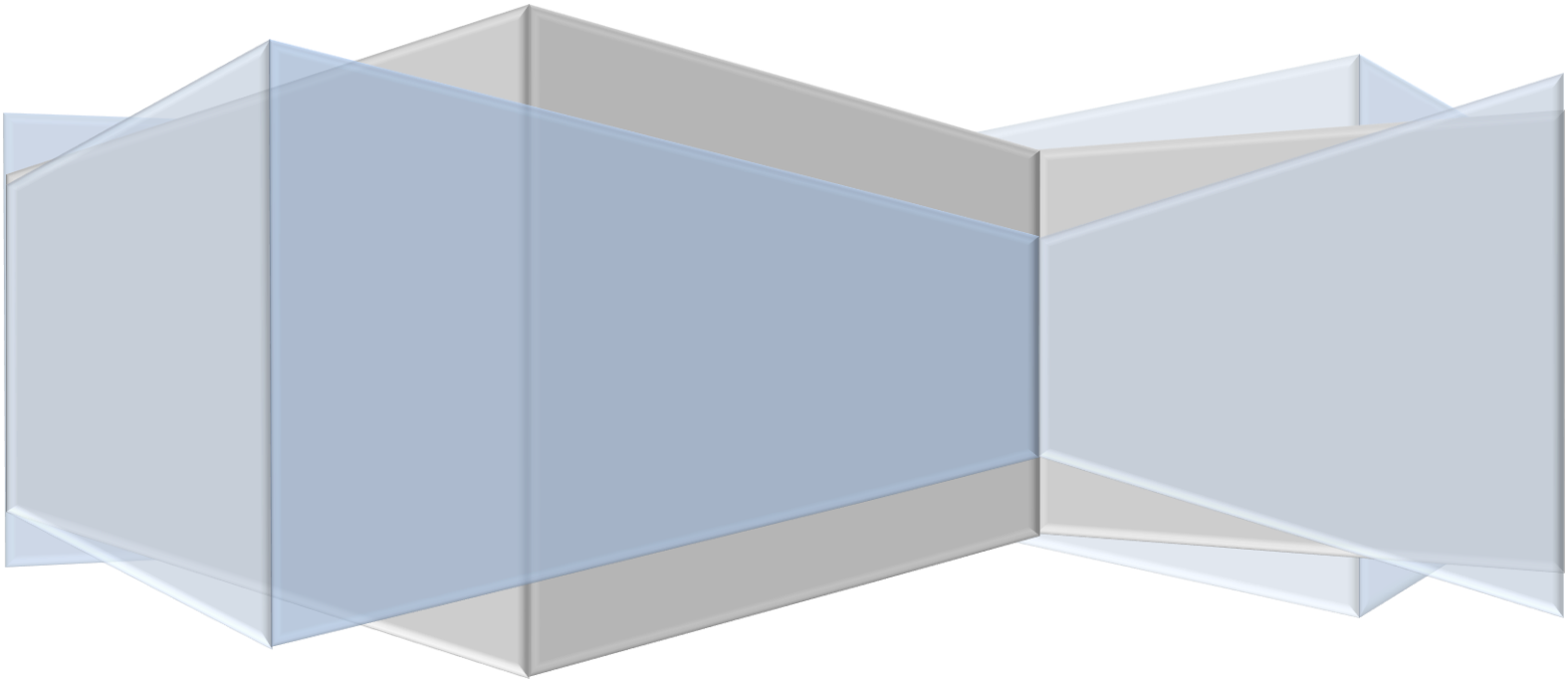
Signature of student and date

Appendix 2: Requirements Specification

Supervisor – Paul Hayes

Student Number: 08399573

Supervisor: Paul Hayes



Requirements Specification (RS)

Introduction

CanvasLauncher is a home replacement application (also known as a launcher) which aims to blend the power of the android devices available today with user focused customization. The reason for this is down to the fact that most phones have a fixed/rigid layout that can merely be modified slightly to other pre-determined designs and settings. Essentially users have to live within the boundaries set out for them by the likes of Samsung, htc and other android device developers.

With the CanvasLauncher this issue will be addressed fully, users of android devices will be free to change and modify the look and feel of their phones from the icon's, animations and themes to the transitions between home screens.

Other launchers on the market are very basic in these areas, and those who excel in them falter in others. It is my aim to build a home replacement application that provides users with robust customization, yet at the same time maintains the high level of performance android users expect.

Purpose

The purpose of this document is to outline and demonstrate both the functional and non-functional requirements of the project. This document is focused on representing what the end user of the application can expect and represents the vision which the project will take over the course of its development.

Project Scope

The scope of the project is to develop an android home replacement application, "android launcher" that is customizable, provides something new in terms of its functionality and design to users and takes advantage of the lack of innovation in this area of android development.

The system shall be built using the android development kit through eclipse. This will be quick and easy to set up and is also free meaning no additional software will need to be purchased to aid in this projects development.

Other programs such as adobe fireworks, paint.net, droid draw etc. will be used over the course of the project for designing and providing templates of the finished application and all of these I have at my disposal cost free.

Definitions, Acronyms, and Abbreviations

- ADK - Android Development Kit
- Android SDK - Android Software Development Kit
- GUI - Graphical User Interface
- Launcher - Home Replacement Application for the Android Platform
- Linux Kernel - this is an operating system used by the linux family, it is an example of free/open source software.

User Requirements Definition

The application will be designed for a wide variety of people as not all those who own an android device are people of a technological background or have an interest in this area.

It is important to firstly state that the primary objective from the user's point of view is to make the application as accessible as possible. The launcher is aimed at being a nice, easy to use home replacement application that allows the user to customize ones phone in terms of its on screen look and design.

Due to this the launcher will;

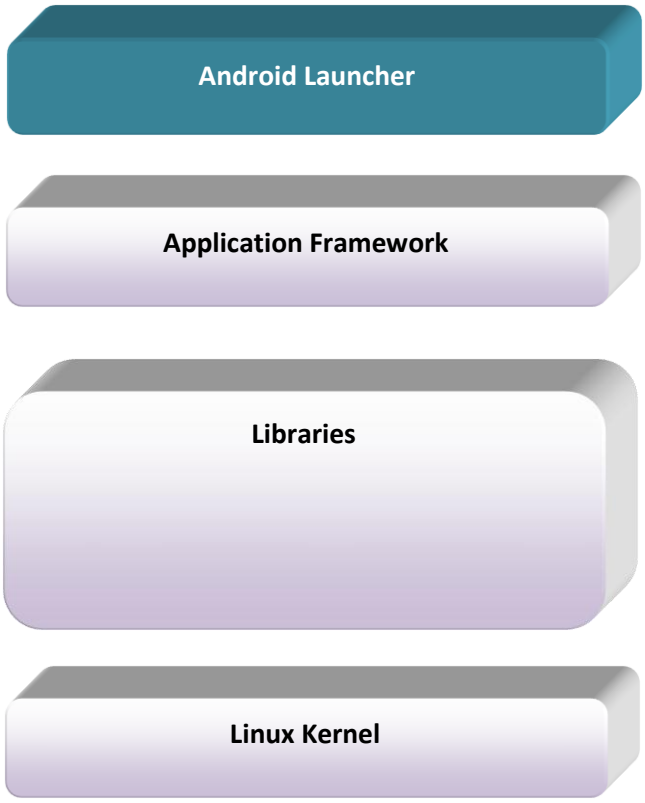
- Have a nice easy to understand layout
- Be easy to use with some tutorials provided to introduce its key features
- Have a simple layout
- Easy to understand icons
- Provide a secondary tab area on the home screen for the most commonly used apps
- The dock display will be customizable
- Super-smooth scrolling
- Animated screen previews
- 2D app drawer
- Up to 7 home screens
- Customizable icons
- Uninstall apps directly from the home screen or app drawer
- Transition effects between home screens
- Unread SMS, Gmail, and missed call badges in the dock icons

System Architecture

The CanvasLauncher's architecture consists of the following;

5. A mobile android device upon which the application will run.
6. The application framework
7. Libraries
8. Linux Kernel

Below is a diagram which shows the applications architecture:



The application will run solely on the android operating system. The GUI will be built using the ADT for eclipse and composed off several images and other content (icons, layouts) composed in fireworks. Other applications such as DroidDraw, AndroidDraw and wireframe templates will also be used in the GUI development.

Interface Requirements

The only interface which is used by the android launcher is the GUI that allows input from the user and then displays an output based on what the user has requested. This GUI will display all menu screens, icons, applications, widgets etc. it is essentially the central hub of all the applications output which is to be displayed to the user.

Description & Priority

This interface displays the icons, home screens, apps and so on to the user and responds to commands by the user via the touch screen. The GUI is the area to which all requests to the system are placed and then returned to the user.

This is essential to the launchers success as without a easy to understand functional GUI the user will not be able to interact with the application, and so the launcher will be useless. For this reason the GUI is a top priority as it is the building blocks upon which the application will be displayed and given functionality.

Physical Environment Requirements

There are no real environmental requirements needed for development of this project. All I will be working with is my home PC and personal laptop upon which the appropriate development software will be installed.

Due to the risk off overheating which may leave the development PC unusable or there being a loss of data I will be keeping both, along with all saved data in a cool area that allows for sufficient ventilation of the systems.

Due to the installation requirements of the ADT I will need;

- Windows 7 (32/64 bit)
- The Eclipse IDE
- JDK 6
- 2GB Disk Space for all installations

Documentation Requirements

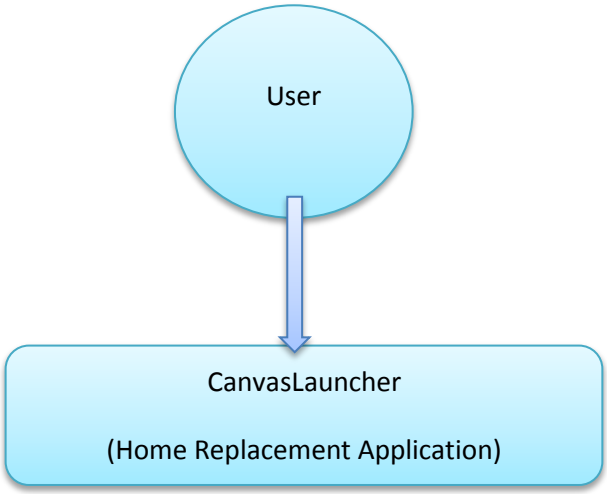
The following is a list of the documents that will be submitted throughout the duration of the project;

1. Project Proposal
2. Requirements Specification
3. Reflective Journals 1 to 7
4. Technical Report

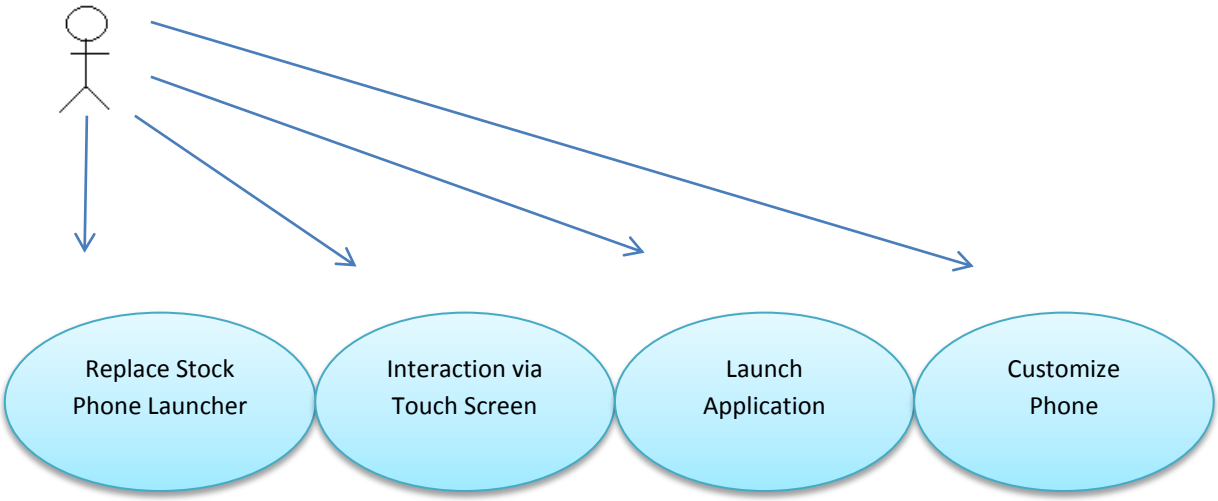
Other documents that are not for submission but are requirements in term of research for the project will mostly come from the android development website and forums.

System Models

Context Diagram:



Use Case Diagram:



Current Issues and Immediate Tasks

Immediate work to be done is the installation of the ADT and android ADK to the Eclipse framework which will give me all the tools I will need for developing the application.

Tasks

1. Android Development
2. GUI
3. Icon/Layout Design
4. Prototype Presentation Preparation
5. Prototype Presentation
6. Report Templates Development
7. Testing

8. Documentation
9. Final Presentation Preparation

Functional Requirements

Requirement 1

The ability to replace the stock android launcher

Description & Priority

This is the most essential requirement as without the application being able to fully replace the stock home launcher the application would not be an efficient home replacement app.

Requirement Activation

This requirement will be activated by the user after the application has been installed. Once installed the application will be ready to run, upon its installation the user will have initialised the application to replace the stock launcher running on the android device.

Technical issues

The main issue in satisfying this requirement is in the range of android devices available, not all devices will run the application smoothly due to certain characteristics as size, firmware, android version and so on. This issue will be taken into account while designing and developing the application to reduce these issues being a prominent concern.

Risks

The circumstances in which this requirement may not be satisfied would be if the device running the application isn't fully compatible due to the constraints mentioned above. If this was to be the case then the home replacement app would not be able to replace the stock launcher and so would be unusable.

Dependencies with other requirements

None.

Functional Requirements

Use Case: Replace Stock Android Launcher

Use Case ID	0001		
Scope	CanvasLauncher Home Replacement Application		
Primary Actor	User		
Brief Description	The use case describes how the user replaces the stock launcher on their android device to the CanvasLauncher		
Preconditions	The user must have an android device that is turned on and 3g or Wi-Fi enabled		
Flow of Events		**Actor Input**	**System Response**
	1	The user downloads the	

		CanvasLauncher via the android market	
	2		The system will download/install the application on the android device.
	3	The user then selects the newly downloaded launcher from their list of downloaded applications	
	4		The system launches the CanvasLauncher which then replaces the stock launcher on the android device.
Post-conditions			
Alternative flows and exceptions	1.a: If the application does not download or install properly the process will have to be repeated in order to get the required result		

Requirement 2

Allows user interaction via the touch screen enabled device.

Description & Priority

This requirement is the bases upon which the user interacts and uses the application, without the application being responsive to the users commands via the touch screen the application will be unusable as it will be unresponsive to user requests.

Requirement Activation

This requirement will be used constantly by the user once the home replacement app has been installed on the android enabled device. The CanvasLauncher will be the way in which the user interacts with their phone to select application such as messaging, email etc. from now on so it is a vitally important function to the applications overall success.

Technical issues

I can't foresee any issue's in this area as there is plenty of online support and references for ensuring your application is touch responsive as it is mandatory requirement for all android applications.

Risks

Issues may arise in implementing this functionality into the application in terms of the applications responsiveness, accuracy and so on. To reduce the risk of this I plan on introducing each of the touch screen actions individually to ensure each one is working appropriately before implementing another.

Dependencies with other requirements

This function is dependent on the device having a touch screen GUI as this is the basis upon which the CanvasLauncher is used by the user and so is a mandatory requirement.

Requirement Activation

The requirement will be used by the user right from the point of activation as it is how the user will interact and command the application.

Functional Requirements

Use Case: Interact via Touch Screen

<table>
<tr><td>Use Case ID</td><td colspan="3">0002</td></tr>
<tr><td>Scope</td><td colspan="3">CanvasLauncher Home Replacement Application</td></tr>
<tr><td>Primary Actor</td><td colspan="3">User</td></tr>
<tr><td>Brief Description</td><td colspan="3">The use case describes how the interacts with the launcher via touch screen commands</td></tr>
<tr><td>Preconditions</td><td colspan="3">The user must have an android device with a touch screen and the CanvasLauncher installed and running on their device</td></tr>
<tr><td rowspan="3">Flow of Events</td><td></td><td>Actor Input</td><td>System Response</td></tr>
<tr><td>1</td><td>The user interacts with the CanvasLauncher via touching/swiping their finger horizontally on the screen</td><td></td></tr>
<tr><td>2</td><td></td><td>The system then provides an output based on the users request</td></tr>
<tr><td>Post-conditions</td><td colspan="3"></td></tr>
<tr><td>Alternative flows and exceptions</td><td colspan="3"></td></tr>
<tr><td></td><td colspan="3"></td></tr>
</table>

Requirement 3

The ability to launch applications installed on the android device.

Description & Priority

As the application is a home replacement app it utilises all the applications installed on the android device and allows the user to continue to use them through its own interface.

Requirement Activation

After the launcher has been installed and is running on the device the user will be shown the applications installed on the phone via the application drawer. Upon selecting an application (for example messages) the CanvasLauncher will begin to run that application normally like the stock launcher would.

Technical issues

The main issue for this requirement is ensuring that the launcher links up with the applications installed on the phone. This is what allows the application to be selected and the phone to be used as it normally would with the stock launcher.

If this does not happen then no application will be able to be launcher via the CanvasLauncher.

Risks

The circumstances in which this requirement may not be satisfied would be if the launcher does not allow for applications to be run when selected. This will make the launcher inefficient and nothing more than a skin with no functionality.

Dependencies with other requirements

This function is dependent on the application fully replacing the stock launcher of the android device.

Functional Requirements

Use Case: Deploying Applications through the Launcher

<table>
<tr><td>Use Case ID</td><td colspan="3">0003</td></tr>
<tr><td>Scope</td><td colspan="3">CanvasLauncher Home Replacement Application</td></tr>
<tr><td>Primary Actor</td><td colspan="3">User</td></tr>
<tr><td>Brief Description</td><td colspan="3">The use case describes how the user can launch applications through the CanvasLauncher</td></tr>
<tr><td>Preconditions</td><td colspan="3">The user must have an android device that is turned on</td></tr>
<tr><td rowspan="5">Flow of Events</td><td></td><td>Actor Input</td><td>System Response</td></tr>
<tr><td>1</td><td>The user opens the application menu via the touch screen</td><td></td></tr>
<tr><td>2</td><td></td><td>The system displays a list of applications the user can then launch</td></tr>
<tr><td>3</td><td>The user then selects their desired application</td><td></td></tr>
<tr><td></td><td></td><td>The system then launches the application through the CanvasLauncher</td></tr>
<tr><td>Post-conditions</td><td colspan="3"></td></tr>
<tr><td>Alternative flows and exceptions</td><td colspan="3"></td></tr>
</table>

Requirement 4

The ability to allow the user to customize the look of their phone

Description & Priority

This requirement is the primary reason for using the application, with allowing users to customize their phone in ways that are not allowed through the stock launcher there is an incentive for using the CanvasLauncher as your home replacement application. It gives users functionality that was previously restricted from them, this can be said to be the main selling point for the launcher so without it there is no need for users to buy and download the CanvasLauncher.

Requirement Activation

The customization features will be accessible via the launcher settings window. Here the user will be able to edit and customize their phone with an array of options.

Technical issues

The main issue in satisfying this requirement is in the range of android devices available, not all devices will run the application the same so allowing each device to avail of each feature may take some time to do.

Risks

The circumstances in which this requirement may not be satisfied would be if the device running the application isn't fully compatible due to the constraints mentioned above. Also implementing each feature may prove to be a time consuming task being loop holes may need to be taken to ensure other areas are given a sufficient amount of time as well.

Dependencies with other requirements

This function is dependent on the application fully replacing the stock launcher of the android device.

Functional Requirements

Use Case: Customizing the Phones Look

<table>
<tr><td>Use Case ID</td><td colspan="3">0004</td></tr>
<tr><td>Scope</td><td colspan="3">CanvasLauncher Home Replacement Application</td></tr>
<tr><td>Primary Actor</td><td colspan="3">User</td></tr>
<tr><td>Brief Description</td><td colspan="3">The use case describes how the user customizes their device using the CanvasLauncher</td></tr>
<tr><td>Preconditions</td><td colspan="3">The user must have an android device that is turned on</td></tr>
<tr><td rowspan="5">Flow of Events</td><td></td><td>Actor Input</td><td>System Response</td></tr>
<tr><td>1</td><td>The user select's preferences from the options menu.</td><td></td></tr>
<tr><td>2</td><td></td><td>The system displays a list of options to do with customizing several areas of the phones look.</td></tr>
<tr><td>3</td><td>The user then selects the area they wish to customize…themes, screen transitions etc.</td><td></td></tr>
<tr><td></td><td></td><td>The system then shows the appropriate options under the</td></tr>
</table>

			selected area.
		The user picks their desired choice	
			The system applies this action to the user's phone.
Post-conditions			
Alternative flows and exceptions			

System Evolution

The system could evolve over time to allow new levels of functionality, new design layouts, improved customizable features, widgets etc. essentially with this being the first version of the launcher there is plenty of room to build and improve upon the initial released design via updates through the android market.

The launchers design is one which has limitless possibilities due to the accessibility of the android format. Essentially due to this any new ideas in terms of functionality and design that benefit the launcher as a whole can be designed and released at ease keeping users happy and engaged with the product.

Conclusions

The functional requirements and user interface are quiet simple in theory, the main focus lies in enabling the application to fully take over and replace the stock launcher already installed on the phone.

With a sufficient amount of time and effort in this area with the resources available this should be accomplished and successfully implemented along with all the other areas necessary for the CanvasLauncher to be a successful launcher.

Appendix 3: Code Segments That Were Unusable

The following code was taken from the Paranoid Android Tutorial on android binding in the hopes it would provide a nice simplistic application drawer for the application. However attempts to use this code were unsuccessful due to the tutorial being a little vague and eclipse not recognising many of the methods and functions in the code.

This Tutorial can be found at: http://andytsui.wordpress.com/2011/07/07/tutorial-android-launcher-with-android-binding-part-0-motivation/

The Code Segments

```
package com.example.applicationpicker;


import android.app.Activity;

import android.os.Binder;

import android.os.Bundle;


public class ApplicationPickerActivity extends Activity {

   /** Called when the activity is first created. */

   @Override

   public void onCreate(Bundle savedInstanceState) {

      super.onCreate(savedInstanceState);

      Binder.setAndBindContentView(this, R.layout.main, new ApplicationPickerViewModel(this));

   }

}


////////////////////////////////////////////////////////////////////////////

package com.example.applicationpicker;
```

```
import java.util.ArrayList;
import java.util.List;
import java.util.Observable;

import android.content.Context;
import android.content.Intent;
import android.content.pm.ResolveInfo;
import android.graphics.drawable.Drawable;

public class ApplicationPickerViewModel {

        private final Context mContext;
        public ApplicationPickerViewModel(Context context){
                mContext = context;
                ApplicationList.addAll(getApplications());
        }

        public final ArrayListObservable
                ApplicationList = new ArrayListObservable();

        private List getApplications(){
                List apps = new ArrayList();

                Intent mainIntent = new Intent(Intent.ACTION_MAIN, null);
    mainIntent.addCategory(Intent.CATEGORY_LAUNCHER);

    List mApps = mContext.getPackageManager().queryIntentActivities(mainIntent, 0);
```

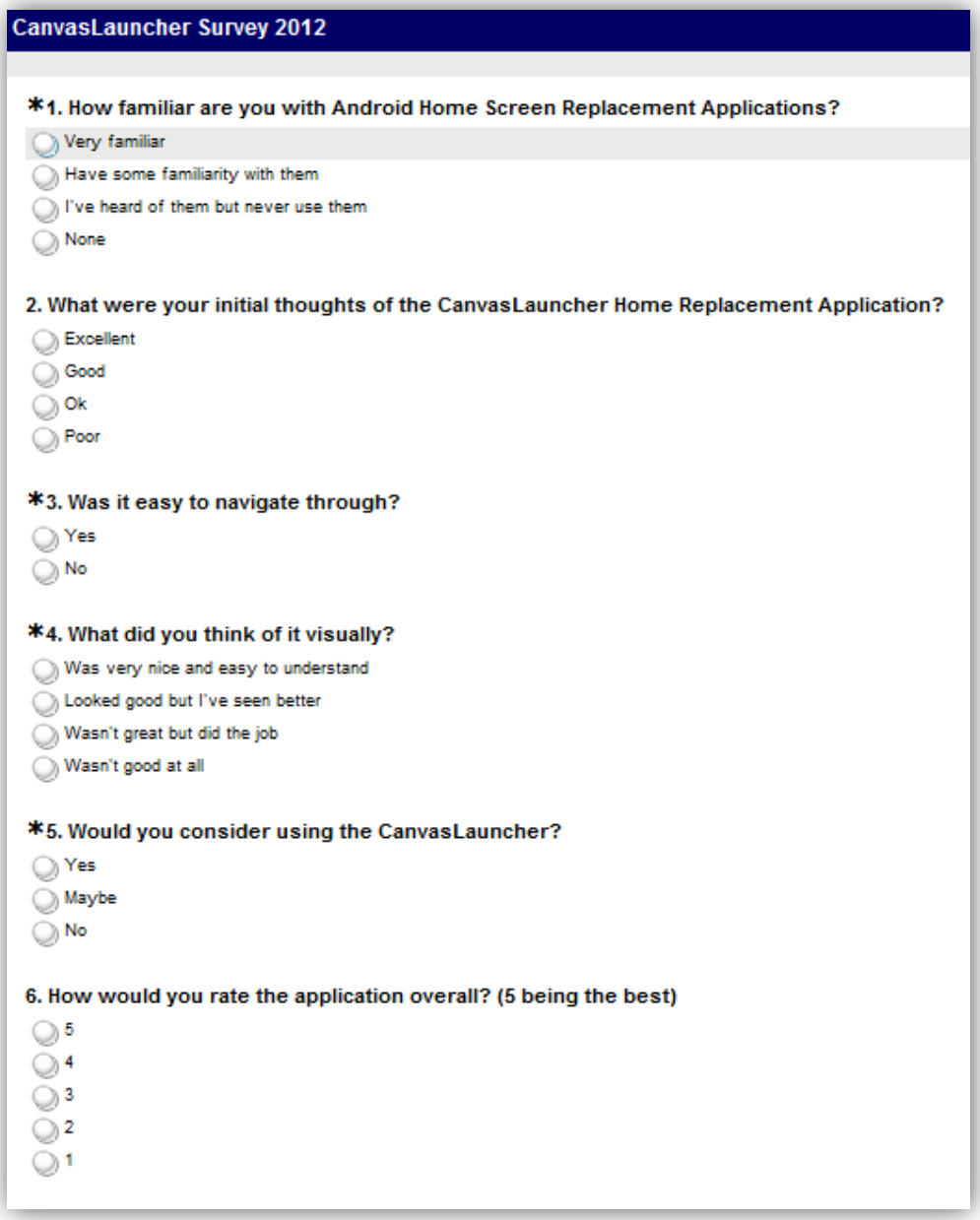
```
int length = mApps.size();
for(int i=0; i ResolveInfo info = mApps.get(i);
    Drawable image = info.loadIcon(mContext.getPackageManager());

    apps.add(new App(
                    info.activityInfo.loadLabel(mContext.getPackageManager()),
                    info.activityInfo.name,
                    image));
}

Collections.sort(apps);

return apps;
    }

    public static class App implements Comparable{
            public final StringObservable Name = new StringObservable();
            public final CharSequenceObservable Label = new CharSequenceObservable();
            public final Observable Icon = new Observable(Drawable.class);

            public App(CharSequence label, String name, Drawable icon) {
                    super();
                    Label.set(label);
                    Name.set(name);
                    Icon.set(icon);
            }
```

```
		public int compareTo(App another) {

			return Name.get().compareTo(another.Name.get());

		}

	}


////////////////////////////////////////////////////////////////////////////

package com.example.applicationpicker;

import java.util.List;

public class ArrayListObservable {

	public void addAll(List applications) {
		// TODO Auto-generated method stub

	}

}


////////////////////////////////////////////////////////////////////////////

package com.example.applicationpicker;

public class CharSequenceObservable {

	public void set(CharSequence label) {
		// TODO Auto-generated method stub

	}

}

////////////////////////////////////////////////////////////////////////////

package com.example.applicationpicker;

public class StringObservable {

	public App get() {
		// TODO Auto-generated method stub
		return null;
	}

	public void set(String name) {
```

```
            // TODO Auto-generated method stub
        }
}
```

Appendix 4: Survey

Below is a copy of the survey used during the testing and evaluation phase of the CanvasLauncher Home Replacement Applications Development.



CanvasLauncher Survey 2012

***1. How familiar are you with Android Home Screen Replacement Applications?**

- Very familiar
- Have some familiarity with them
- I've heard of them but never use them
- None

2. What were your initial thoughts of the CanvasLauncher Home Replacement Application?

- Excellent
- Good
- Ok
- Poor

***3. Was it easy to navigate through?**

- Yes
- No

***4. What did you think of it visually?**

- Was very nice and easy to understand
- Looked good but I've seen better
- Wasn't great but did the job
- Wasn't good at all

***5. Would you consider using the CanvasLauncher?**

- Yes
- Maybe
- No

6. How would you rate the application overall? (5 being the best)

- 5
- 4
- 3
- 2
- 1

4.1 Survey Findings

Below are the results of the user survey.

1. How familiar are you with Android Home Screen Replacement Applications? Create Chart Download

	Response Percent	Response Count
Very familiar	27.3%	6
Have some familiarity with them	4.5%	1
I've heard of them but never use them	40.9%	9
None	27.3%	6
answered question		22
skipped question		0

2. What were your initial thoughts of the CanvasLauncher Home Replacement Application? Create Chart Download

	Response Percent	Response Count
Excellent	27.3%	6
Good	45.5%	10
Ok	22.7%	5
Poor	4.5%	1
answered question		22
skipped question		0

3. Was it easy to navigate through? Create Chart Download

	Response Percent	Response Count
Yes	95.5%	21
No	4.5%	1
answered question		22
skipped question		0

4. What did you think of it visually? Create Chart Download

	Response Percent	Response Count
Was very nice and easy to understand	59.1%	13
Looked good but I've seen better	27.3%	6
Wasn't great but did the job	9.1%	2
Wasn't good at all	4.5%	1
answered question		22
skipped question		0

5. Would you consider using the CanvasLauncher? Create Chart Download

	Response Percent	Response Count
Yes	45.5%	10
Maybe	40.9%	9
No	13.6%	3
answered question		22
skipped question		0

6. How would you rate the application overall? (5 being the best) Create Chart Download

	Response Percent	Response Count
5	27.3%	6
4	45.5%	10
3	22.7%	5
2	0.0%	0
1	4.5%	1
answered question		22
skipped question		0

7. Any comments you may have about the CanvasLauncher please feel free to write below Download

	Response Count
Show Responses	7
answered question	7
skipped question	15

Appendix 5: Reflective Journals

[Reflective Journal

Created by Colin Greene – x08399573

Colin Greene

08399573

Software Systems Year 4

Android Launcher (Home Replacement Application)

REFLECTIVE JOURNAL

Over the last month I have been laying down the foundations upon which the rest of the project will follow. This has involved;

- Creating and handing in a project proposal document stating the overview of my android launcher.
- Writing up the requirements specification document describing the project in detail from its layout and architecture to its functionality.
- Setting up my development PC with the required software for developing my application, the following were installed:
    - Eclipse
    - Android SDK
    - Fireworks

Other work done has involved doing basic tutorials to familiarize myself with the android environment along with developing icons, layout designs and so on that will be improved upon in time as the application expands and takes a more permanent shape.

The only real issue that has arose during this time was during the installation of the Eclipse IDE, this proved somewhat troublesome as it would not recognize the java files installed on the PC which it needed in order to run. After messing around with this issue the problem was identified and fixed, the problem arose due to the location folder in which Eclipse was being installed, after changing this the problem was rectified.

[Reflective Journal

Created by Colin Greene – x08399573

Reflective Journal

Since my last journal entry I have been continuing working with online tutorials and guides. Some are proving more helpful than others but due to the lack of overall online resources in relation to the topic of home screen application development some issues have arose.

Due to this lack of resources I feel that the development of some functionality and design features may be effected in terms of their quality as more time is needed to be invested into understanding and developing a comprehensive home screen replacement application.

This has been the course of action I've been taking over the last few weeks along with keeping documentation on the development process for the technical report as progress is being made since keeping track of this progress is important.

Reflective Journal

The following is a summary of the tasks, problems, issues and relevant solutions which have come to the forefront during developing my BSHSS4 Software Project.

Created by Colin Greene – x08399573

Reflective Journal

Since my last reflective journal I have become the core development of my android launcher. So far I have an app drawer working which pulls every application from my phone into the launcher's interface. Also I can store the 3 most recent applications used by the user on the home screen dock bar.

In conjunction to this I have a small widget application working which the user can set on any home screen, it is simply a visual widget as I have yet to implement any functionality into it but I hope to sort this out over the coming weeks.

The main problem I've encountered has been with the setting wallpaper functionality on the launcher, it works well on the galaxy s 2 and emulator but on the galaxy ace there are some bugs which have made it fairly temper mental

Other than the above 3 things I have nothing else to report in relation to the development of my android launcher.

[Reflective Journal

Created by Colin Greene – x08399573

Reflective Journal

Over the last while I've continued working on the progress I made up to the point of the midterm presentation. I've been mainly focusing on the home replacement application and trying to fix and tidy it up a little so it looks more presentable for the end user.

Also I have begun to look at some different testing methods so I can evaluate my project in the best way possible.

Again the main issue has been with making solid advancements with the project due to the lack of online guides, tutorials etc along with other resources that are related to the area of home screen replacements. This has caused some delays with other aspects such as theme design and icon design but I do hope to make sufficient room for them in the coming weeks.

[Reflective Journal

Created by Colin Greene – x08399573

Colin Greene

08399573

Software Systems Year 4

Android Launcher (Home Replacement Application)

Reflective Journal

Issues have been arising quiet a lot lately as implementing functionality and other add on features have forced the application to break or not run correctly when installed on a device or ran on the Eclipse IDE. This has led me to use the method of unit testing just to ensure everything is running as expecting independently. This does not ensure everything will run smoothly when joined together in the later stages of development but for the moment it is the only way forward in terms of making distinguishable progress.

Other notable areas are in relation to the theme design, I have found one or two guides to this online but have yet to research them properly as of yet, in relation to this also I have looked further into some other design features such as widget styles and icon styles.

The documentation is still continually being updated just to ensure that nothing is forgotten.

[Reflective Journal

Created by Colin Greene – x08399573

Reflective Journal

No real significant news to report for this month. The same issues and process of development as before are still at the forefront of the project. With other projects in distributed systems 2 and artificial intelligence along with a report for strategic management the main project in recent weeks has taking a bit of a back seat. These modules have taken a priority over the project as they are all due over the coming weeks and need a sufficient amount of time to be invested in them.

[Reflective Journal

Created by Colin Greene – x08399573

REFLECTIVE JOURNAL

Work on the project has continued but again at a slower rate than in earlier months. This is due to a combination of technical and development issues along with other priorities, which in this case where my final year exams. These took place mid-April this year which was a surprise as all other end of year exams have taken place in May, this caused me to have to adjust my project plans to allow for a sufficient amount of time to study.

The other issues still remain;

1. Problems with implementing functionality
2. Lack of resources for specific topics
3. Eclipse and ADT updates causing working application's to no longer run as before

Work for the documentation has continued and this will be fixed up and prepared for submission and late editions such as user acceptance testing results over the coming weeks.

This is all I can think of that needs to be reported at this time.